B. Amendments to the Claims:

Claim 1 (Currently Amended): A method for enabling an application designer and a user to develop a User Interface (UI) from a plurality of UI models without coding, each of the plurality of UI models being developed using a plurality of pre-built reusable components, the method comprising the steps of:

a. identifying the requirements of the UI as processes, the application designer identifying the processes;

b. defining tasks that are required to define the identified processes, the tasks being defined by providing meta data to instances of a set of pre-built reusable components, the meta-data being provided using a visual modeling environment, each pre-built reusable component being an abstract object built to perform a function;

c. enabling the application designer to verify the defined tasks by applying a set of pre-defined verifications on each of the defined tasks;

[[c]] d. connecting the defined tasks in a logical order using the visual modeling environment, the defined tasks being connected to model the identified processes, the identified processes being used to develop the plurality of UI models;

[[d]] e. storing the plurality of UI models in a database; ~~and~~

[[e]] f. executing the plurality of UI models by an engine based on a plurality of requests, the plurality of requests being made by the application designer and the user while developing the UI, the plurality of requests being concurrently executed by loading the identified processes from the database and executing the tasks specified in the identified processes, the tasks being executed in the logical order[[.]] ; and

g. enabling the application designer to visually verify the developed UI, the verification comprising the steps of:

i. observing values of a plurality of watch variables while executing the plurality of UI models, the plurality of watch variables being identified by the application designer;

ii. stopping at each of a plurality of break points while executing the plurality of UI models, the plurality of break points being set by the application designer; and

iii. analyzing information related to the plurality of UI models at each of the plurality of break points.

Claim 2 (Cancelled).

Claim 3 (Previously Presented): The method according to claim 1, wherein the step of defining the tasks comprises:

a. defining the tasks that are required to define the processes that are required to develop UI screens comprising:

i. defining the tasks for placing controls on the UI screens;

ii. defining the tasks for mapping of database fields to screen variables;

iii. defining the tasks for validating user actions at a control level and at a screen level;

iv. defining the tasks for forwarding screen information to an application controller; and

b. defining the tasks that are required to define the processes that are required to develop the structure of the UI.

Claim 4 (Previously Presented): The method according to claim 3, wherein the step of defining the tasks comprises the steps of:

a. selecting the set of pre-built reusable components that are required to define the tasks; and

b. defining the tasks, the tasks being defined by specifying meta-data for the selected pre-built reusable components, the meta-data being process specific properties which when associated with a pre-built reusable component defines the corresponding task.

Claim 5 (Previously Presented): The method according to claim 1, wherein the step of connecting the defined tasks in the logical order comprises the steps of:

a. defining at least one Rule;

b. specifying tasks to be referred to in case of success of the at least one Rule; and

c. specifying tasks to be referred to in case of failure of the at least one Rule.

Claims 6-10 (Cancelled).

Claim 11 (Previously Presented): The method according to claim 1, wherein the step of executing the plurality of UI models comprises the steps of:

a. inputting the plurality of requests to be processed, the plurality of requests being input by the application designer and the user while developing the UI;

b. transferring the plurality of requests to the Engine for processing;

c. identifying the pre-built reusable components required to process the plurality of requests;

d. caching the identified pre-built reusable components;

e. executing the tasks in the logical order as defined in the identified processes;

f. handling errors that occur while processing the plurality of requests, if errors occur in any of the above steps;

g. logging the information related to the execution of tasks in the database; and

h. outputting the results of the execution.

Claim 12 (Currently Amended): A method for enabling an application designer and a user to develop a User Interface (UI) from a plurality of UI models without coding, each of the plurality of UI models being developed using a plurality of pre-built reusable components, the method comprising the steps of:

a. identifying the requirements of the UI as processes, the application designer identifying the processes;

b. defining tasks that are required to define the identified processes, the tasks being defined by providing meta-data to instances of a set of pre-built reusable components, the meta-data being provided using a visual modeling environment, each pre-built reusable component being an abstract object built to perform a function;

c. verifying the defined tasks by applying a set of pre-defined verifications on each of the defined task;

d. connecting the defined tasks in a logical order using the visual modeling environment, the defined tasks being connected to model the identified processes, the identified processes being used to develop the plurality of UI models;

e. storing the plurality of UI models in a database;

f. inputting a plurality of requests to be processed, the plurality of requests being input by the application designer and the user while developing the UI;

g. transferring the plurality of requests to an Engine for processing;

h. identifying the pre-built reusable components required to process the plurality of requests;

i. caching the identified pre-built reusable components;

j. executing the tasks in the logical order as defined in the identified processes;

k. enabling the application designer to visually verify the developed UI, the verification comprising the steps of:

i. observing values of a plurality of watch variables while executing the plurality of UI models, the plurality of watch variables being identified by the application designer;

ii. stopping at each of a plurality of break points while executing the plurality of UI models, the plurality of break points being set by the application designer; and

iii. analyzing information related to the plurality of UI models at each of the plurality of break points;

[[k]] l. handling errors that occur while processing the plurality of requests, if errors occur in any of the above steps;

[[l]] m. logging the information related to the execution of tasks in the database; and

[[m]] n. outputting the results of the execution.

Claims 13-23 (Cancelled)

Claim 24 (Currently Amended): A computer program product for use with a computer, the computer program product comprising a computer usable medium having a computer readable program code embodied therein for enabling an application designer and a user to develop a User Interface (UI) from a plurality of UI models without coding, each of the plurality of UI models being developed using a plurality of pre-built reusable components, the computer program code performing the steps of:

a. identifying the requirements of the UI as processes, the application designer identifying the processes;

b. defining tasks that are required to define the identified processes, the tasks being defined by providing meta-data to instances of a set of pre-built reusable components, the meta-data being provided using a visual modeling environment, each pre-built reusable component being an abstract object built to perform a function;

c. enabling the application designer to verify the defined tasks by applying a set of pre-defined verifications on each of the defined tasks;

[[c]] d. connecting the defined tasks in a logical order using the visual modeling environment, the defined tasks being connected to model the identified processes, the identified processes being used to develop the plurality of UI models;

[[d]] e. storing the plurality of UI models in a database; ~~and~~

[[e]] f. executing the plurality of UI models by an engine based on a plurality of requests, the plurality of requests being made by the application designer and the user while developing the UI, the plurality of requests being concurrently executed by loading the identified processes from the database and executing the tasks specified in the identified processes, the tasks being executed in the logical order[[.]] ; and

g. enabling the application designer to visually verify the developed UI, the verification comprising the steps of:

i. observing values of a plurality of watch variables while executing the plurality of UI models, the plurality of watch variables being identified by the application designer;

ii. stopping at each of a plurality of break points while executing the plurality of UI models, the plurality of break points being set by the application designer; and

iii. analyzing information related to the plurality of UI models at each of the plurality of break points.

Claim 25 (Cancelled).

Claim 26 (Previously Presented): The computer program product as claimed in claim 24, the computer program product further comprising program code embodied therein for enabling the application designer and the user to execute the plurality of UI models using the computer program product, the computer program code performing the steps of:

a. inputting the plurality of requests to be processed, the plurality of requests being input by the application designer and the user while developing the UI;

b. transferring the plurality of requests to the Engine for processing;

c. identifying the pre-built reusable components required to process the plurality of requests;

d. caching the identified pre-built reusable components;

e. executing the tasks in the logical order as defined in the identified processes;

f. handling errors that occur while processing the plurality of requests, if errors occur in any of the above steps;

g. logging the information related to the execution of the tasks in the database; and

h. outputting the results of the execution.

Claim 27-28 (Cancelled):